

Uso de Sentencias DDL para Crear y Gestionar Tablas

ORACLE

Objetivos

Al finalizar esta lección, debería estar capacitado para:

- **Clasificar los objetos principales de la base de datos**
- **Revisar la estructura de tabla**
- **Mostrar los tipos de datos que están disponibles para las columnas**
- **Crear una tabla simple**
- **Comprender cómo se crean las restricciones en el momento de la creación de la tabla**
- **Describir cómo funcionan los objetos de esquema**

Objetos de Base de Datos

Objeto	Descripción
Tabla	Unidad básica de almacenamiento; compuesta por filas
Vista	Representa de forma lógica subconjuntos de datos de una o más tablas
Secuencia	Genera valores numéricos
Índice	Mejora el rendimiento de algunas consultas
Sinónimo	Proporciona nombres alternativos a objetos

Reglas de Nomenclatura

Los nombres de tablas y de columnas:

- **Deben comenzar por una letra**
- **Deben tener una longitud de 1 a 30 caracteres**
- **Sólo deben contener A–Z, a–z, 0–9, _, \$ y #**
- **No deben duplicar el nombre de otro objeto que sea propiedad del mismo usuario**
- **No deben ser palabras reservadas de Oracle Server**

La Sentencia CREATE TABLE

- **Debe tener:**
 - El privilegio CREATE TABLE
 - Un área de almacenamiento

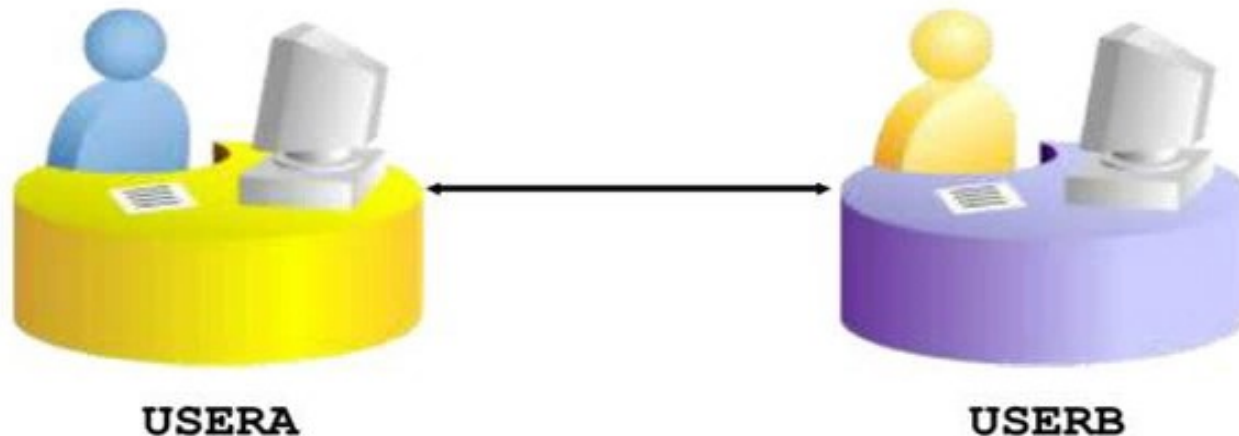
```
CREATE TABLE [schema.]table  
      (column datatype [DEFAULT expr] [, ...]);
```

- **Se especifica:**
 - Nombre de tabla
 - Nombre de columna, tipo de datos de columna y tamaño de columna



Referencia a Tablas de Otro Usuario

- Las tablas pertenecientes a otros usuarios no están en el esquema del usuario.
- Debe utilizar el nombre del propietario como prefijo de dichas tablas.



`SELECT *`
`FROM userB.employees;`

`SELECT *`
`FROM userA.employees;`

Opción DEFAULT

- **Especifica un valor por defecto para una columna durante una inserción.**

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- **Los valores literales, las expresiones o las funciones SQL son valores legales.**
- **El nombre de otra columna o una pseudocolumna son valores ilegales.**
- **El tipo de datos por defecto debe corresponder con el tipo de datos de la columna.**

```
CREATE TABLE hire_dates  
  (id          NUMBER(8),  
   hire date DATE DEFAULT SYSDATE);
```

Table created.

Creación de Tablas

- Cree la tabla.

```
CREATE TABLE dept
  (deptno      NUMBER(2) ,
   dname       VARCHAR2(14) ,
   loc         VARCHAR2(13) ,
   create_date DATE DEFAULT SYSDATE) ;
```

Table created.

- Confirme la creación de la tabla.

```
DESCRIBE dept
```

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)
CREATE_DATE		DATE

Tipos de Datos

Tipo de Datos	Descripción
<code>VARCHAR2 (size)</code>	Datos de carácter de longitud variable;
<code>CHAR (size)</code>	Datos de carácter de longitud fija
<code>NUMBER (p, s)</code>	Datos numéricos de longitud variable
<code>DATE</code>	Valores de fecha y hora
<code>LONG</code>	Datos de caracteres de longitud variable (hasta 2 GB)
<code>CLOB</code>	Datos de caracteres (hasta 4 GB)
<code>RAW y LONG RAW</code>	Datos RAW binarios
<code>BLOB</code>	Datos binarios (hasta 4 GB)
<code>BFILE</code>	Datos binarios almacenados en un archivo externo (hasta 4 GB)
<code>ROWID</code>	Sistema numérico de base 64 que representa la única dirección de una fila en su tabla

Tipos de Datos de Fecha/Hora

Puede utilizar varios tipos de datos de fecha/hora:

Tipo de Datos	Descripción
TIMESTAMP	Fecha con fracciones de segundos
INTERVAL YEAR TO MONTH	Se almacena como intervalo de años y meses
INTERVAL DAY TO SECOND	Se almacena como intervalo de días, horas, minutos y segundos



Tipos de Datos de Fecha/Hora

- El tipo de datos **TIMESTAMP** es una extensión del tipo de datos **DATE**.
- Almacena el año, el mes y el día del tipo de datos **DATE**, más el valor de hora, minutos y segundos, así como el valor de fracciones de segundos.
- Opcionalmente, puede especificar la zona horaria.

```
TIMESTAMP[ (fractional_seconds_precision) ]
```

```
TIMESTAMP[ (fractional_seconds_precision) ]  
WITH TIME ZONE
```

```
TIMESTAMP[ (fractional_seconds_precision) ]  
WITH LOCAL TIME ZONE
```


Tipos de Datos de Fecha/Hora

- El tipo de datos **INTERVAL YEAR TO MONTH** almacena un período de tiempo mediante los campos **YEAR** y **MONTH**:

```
INTERVAL YEAR [(year_precision)] TO MONTH
```

- El tipo de datos **INTERVAL DAY TO SECOND** almacena un período de tiempo en días, horas minutos y segundos:

```
INTERVAL DAY [(day_precision)]  
TO SECOND [(fractional_seconds_precision)]
```


Tipo de Datos INTERVAL DAY TO SECOND

INTERVAL DAY TO SECOND almacena un período de tiempo en días, horas minutos y segundos.

```
INTERVAL '4 5:12:10.222' DAY TO SECOND(3)
```

Indica 4 días, 5 horas y 12 minutos, 10 segundos y 222 milésimas de segundo.

```
INTERVAL '4 5:12' DAY TO MINUTE
```

Indica 4 días, 5 horas y 12 minutos.

```
INTERVAL '400 5' DAY(3) TO HOUR
```

Indica 400 días y 5 horas.

```
INTERVAL '11:12:10.2222222' HOUR TO SECOND(7)
```

Indica 11 horas, 12 minutos y 10,2222222 segundos.

Inclusión de Restricciones

- Las restricciones fuerzan las reglas en el nivel de tabla.
- Las restricciones evitan el borrado de una tabla si hay dependencias.
- Son válidos los siguientes tipos de restricciones:
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK



Instrucciones de Restricciones

- Puede nombrar una restricción u Oracle Server generará un nombre con el formato SYS_Cn.
- Cree una restricción en alguno de estos momentos:
 - Al mismo tiempo que se crea la tabla
 - Después de crear la tabla
- Defina una restricción de nivel de columna o de tabla.
- Visualice una restricción en el diccionario de datos.

Definición de Restricciones

- **Sintaxis:**

```
CREATE TABLE [schema.] table  
    (column datatype [DEFAULT expr]  
    [column_constraint],  
    ...  
    [table_constraint] [, ...] );
```

- **Restricción de nivel de columna:**

```
column [CONSTRAINT constraint_name] constraint_type,
```

- **Restricción de nivel de tabla:**

```
column, ...  
[CONSTRAINT constraint_name] constraint_type  
(column, ...),
```


Definición de Restricciones

- **Restricción de nivel de columna:**

```
CREATE TABLE employees (  
  employee_id  NUMBER(6)  
    CONSTRAINT emp_emp_id_pk PRIMARY KEY,  
  first_name   VARCHAR2(20),  
  ...);
```

1

- **Restricción de nivel de tabla:**

```
CREATE TABLE employees (  
  employee_id  NUMBER(6),  
  first_name   VARCHAR2(20),  
  ...  
  job_id       VARCHAR2(10) NOT NULL,  
  CONSTRAINT emp_emp_id_pk  
    PRIMARY KEY (EMPLOYEE_ID));
```

2

Restricción NOT NULL

Garantiza que no se permitan valores nulos en la columna:

EMPLOYEE_ID	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	90
101	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	90
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	90
103	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	60
104	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	60
178	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	
200	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	10

...
20 rows selected.

↑
Restricción NOT NULL
(Ninguna fila puede
contener un valor
nulo para esta
columna.)

↑
**Restricción
NOT NULL**

↑
**Ausencia de
restricción NOT NULL**
(Cualquier fila puede
contener un valor nulo
para esta columna.)

ORACLE

Restricción UNIQUE

Restricción UNIQUE

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	EMAIL
100	King	SKING
101	Kochhar	NKOCHHAR
102	De Haan	LDEHAAN
103	Hunold	AHUNOLD
104	Ernst	BERNST

...

↑ INSERT INTO

208	Smith	JSMITH
209	Smith	JSMITH

← Permitido
← No permitido:
ya existe

ORACLE

Restricción UNIQUE


Se define en los niveles de tabla o de columna:

```
CREATE TABLE employees (  
  employee_id      NUMBER(6),  
  last_name        VARCHAR2(25) NOT NULL,  
  email            VARCHAR2(25),  
  salary           NUMBER(8,2),  
  commission_pct   NUMBER(2,2),  
  hire_date        DATE NOT NULL,  
  ...  
  CONSTRAINT emp_email_uk UNIQUE(email));
```


Restricción PRIMARY KEY

DEPARTMENTS

PRIMARY KEY



DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

...

No permitido
(valor nulo)



INSERT INTO



	Public Accounting		1400
50	Finance	124	1500

No permitido
(50 ya existe)

ORACLE

Restricción FOREIGN KEY

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

**PRIMARY
KEY** →

...

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
107	Lorentz	60

← **FOREIGN
KEY**

...



INSERT INTO

200	Ford	9
201	Ford	60

**No permitido
(9 no existe)**

← **Permitido**

ORACLE

Restricción FOREIGN KEY

Se define en los niveles de tabla o de columna:

```
CREATE TABLE employees(  
  employee_id      NUMBER(6),  
  last_name        VARCHAR2(25) NOT NULL,  
  email            VARCHAR2(25),  
  salary           NUMBER(8,2),  
  commission_pct   NUMBER(2,2),  
  hire_date        DATE NOT NULL,  
  ...  
  department_id    NUMBER(4),  
  CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)  
    REFERENCES departments(department_id),  
  CONSTRAINT emp_email_uk UNIQUE(email));
```

Restricción FOREIGN KEY: Palabras Clave

- **FOREIGN KEY:** Define la columna de la tabla secundaria en el nivel de restricción de tabla
- **REFERENCES:** Identifica la tabla y la columna de la tabla principal
- **ON DELETE CASCADE:** Suprime las filas dependientes de la tabla secundaria si se suprime una fila de la tabla principal
- **ON DELETE SET NULL:** Convierte valores clave de clave ajena dependientes en valores nulos

Restricción CHECK

- Define una condición que debe satisfacer cada fila
- No se permiten las expresiones siguientes:
 - Referencias a las pseudocolumnas CURRVAL, NEXTVAL, LEVEL y ROWNUM
 - Llamadas a las funciones SYSDATE, UID, USER y USERENV
 - Consultas que hagan referencia a otros valores de otras filas

```
..., salary NUMBER(2)  
CONSTRAINT emp_salary_min  
CHECK (salary > 0),...
```

CREATE TABLE: Ejemplo

```
CREATE TABLE employees
( employee_id      NUMBER(6)
  CONSTRAINT emp_employee_id PRIMARY KEY
, first_name       VARCHAR2(20)
, last_name        VARCHAR2(25)
  CONSTRAINT emp_last_name_nn NOT NULL
, email            VARCHAR2(25)
  CONSTRAINT emp_email_nn    NOT NULL
  CONSTRAINT emp_email_uk    UNIQUE
, phone_number     VARCHAR2(20)
, hire_date        DATE
  CONSTRAINT emp_hire_date_nn NOT NULL
, job_id           VARCHAR2(10)
  CONSTRAINT emp_job_nn      NOT NULL
, salary           NUMBER(8,2)
  CONSTRAINT emp_salary_ck   CHECK (salary>0)
, commission_pct   NUMBER(2,2)
, manager_id       NUMBER(6)
, department_id    NUMBER(4)
  CONSTRAINT emp_dept_fk     REFERENCES
    departments (department_id));
```

Violación de Restricciones

```
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110;
```

```
UPDATE employees
      *
ERROR at line 1:
ORA-02291: integrity constraint (HR.EMP_DEPT_FK)
violated - parent key not found
```

El departamento 55 no existe.

Violación de Restricciones

No se puede suprimir una fila que contenga una clave primaria que se utilice como clave ajena en otra tabla.

```
DELETE FROM departments
WHERE      department_id = 60;
```

```
DELETE FROM departments
      *
ERROR at line 1:
ORA-02292: integrity constraint (HR.EMP_DEPT_FK)
violated - child record found
```


Creación de una Tabla mediante una Subconsulta

- Cree una tabla e inserte filas combinando la sentencia **CREATE TABLE** y la opción **AS subquery**.

```
CREATE TABLE table  
            [(column, column...)]  
AS subquery;
```

- Asigne el número de columnas especificadas al número de columnas de subconsulta.
- Defina columnas con nombres de columna y valores por defecto.

Creación de una Tabla mediante una Subconsulta

```
CREATE TABLE dept80
AS
  SELECT  employee_id, last_name,
          salary*12 ANNSAL,
          hire_date
  FROM    employees
  WHERE   department_id = 80;
```

Table created.

```
DESCRIBE dept80
```

Name	Null?	Type
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

Sentencia ALTER TABLE

Utilice la sentencia ALTER TABLE para:

- **Agregar una nueva columna**
- **Modificar una columna existente**
- **Definir un valor por defecto para la nueva columna**
- **Borrar una columna**

Borrado de una Tabla

- Se suprimen todos los datos y la estructura de la tabla.
- Se validan todas las transacciones pendientes.
- Se borran todos los índices.
- Se borran todas las restricciones.
- *No se puede* hacer rollback de la sentencia DROP TABLE.

```
DROP TABLE dept80;  
Table dropped.
```


Resumen

En esta lección, ha aprendido a utilizar la sentencia `CREATE TABLE` para crear una tabla e incluir restricciones.

- **Clasificar los objetos principales de la base de datos**
- **Revisar la estructura de tabla**
- **Mostrar los tipos de datos que están disponibles para las columnas**
- **Crear una tabla simple**
- **Comprender cómo se crean las restricciones en el momento de la creación de la tabla**
- **Describir cómo funcionan los objetos de esquema**

Práctica 9: Visión General

Esta práctica cubre los temas siguientes:

- **Creación de nuevas tablas**
- **Creación de una nueva tabla mediante la sintaxis
CREATE TABLE AS**
- **Verificación de que las tablas existen**
- **Borrado de tablas**